



Software Communities at ESIP via Open Source

Chris A. Mattmann

Senior Computer Scientist, NASA Jet Propulsion Laboratory

Adjunct Assistant Professor, Univ. of Southern California

Member, Apache Software Foundation



Roadmap

- Areas for Open Source within the ESIP context
 - Strategies and Decision Points
 - Discuss outcome from the NASA OSS
 - Discussion and Recommendations
-
- This is a shorter version of a longer, **half day breakout on NASA Open Source** that I led at the 9th NASA Earth Science Data System Working Group Meetings in October 2010
 - <http://s.apache.org/anM>
 - Also check out **NASA OSS** slides that inspired this: <http://s.apache.org/pw>
 - **ESIP Summer 2011** Session: <http://s.apache.org/Wd0>



And you are?

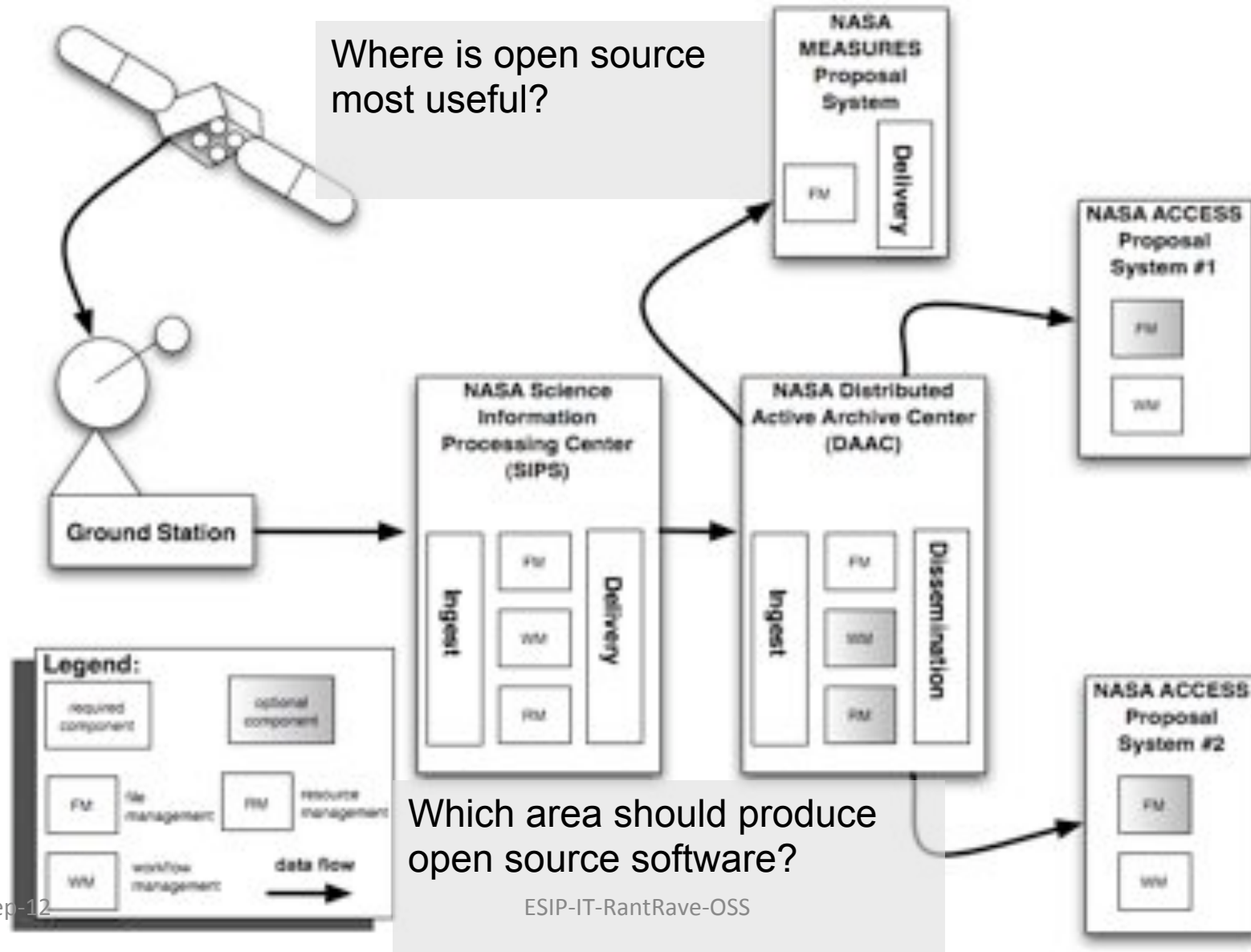


- Senior Computer Scientist at NASA JPL in Pasadena, CA USA
- Software Architecture/Engineering Prof at Univ. of Southern California

- Apache Member involved in
 - OODT (VP, PMC), Tika (PMC), Nutch (PMC), Incubator (PMC), Gora (PMC), SIS (Mentor), cTAKES (Mentor), Airavata (Mentor)



NASA Earth Science DS Context





Concerns in the Open Source World

- Licensing
 - GPL(v2, v3?), LGPL(v?), BSD, MIT, ASLv2
 - Your own custom license approved by OSS
 - NASA OSS license?
 - Caltech license?
 - Copy-left versus Copy-right
- Redistribution
 - Can you take open source product X and use it in your commercially interested software Y?
 - If so, do you have to pay for it?
 - Should others pay for your open source product if they use it in their commercial application?
- Open Source “Help Desk” Syndrome versus Community
 - Are you trying to simply make your open source software (releases) available for distribution (aka *help desk*)?
 - Are you trying to get others to “buy in” to your open source software?



Concerns in the Open Source World

- Intellectual Property
 - Who owns it?
 - How does the Open Source Software affect your IP?
- Open Source Ecosystems
 - Where can you find the “killer app” you need?
 - Which communities are conducive for longevity?
 - How relevant are “generic” open source software communities to NASA Earth Science Data Systems?
- Contributing
 - Are you *even allowed* to contribute to a OSS community?
 - Can you do it on “company” time?
 - What’s required?
 - What’s the governance?
- Responsiveness
 - How response is the OSS community to your projects’ needs?

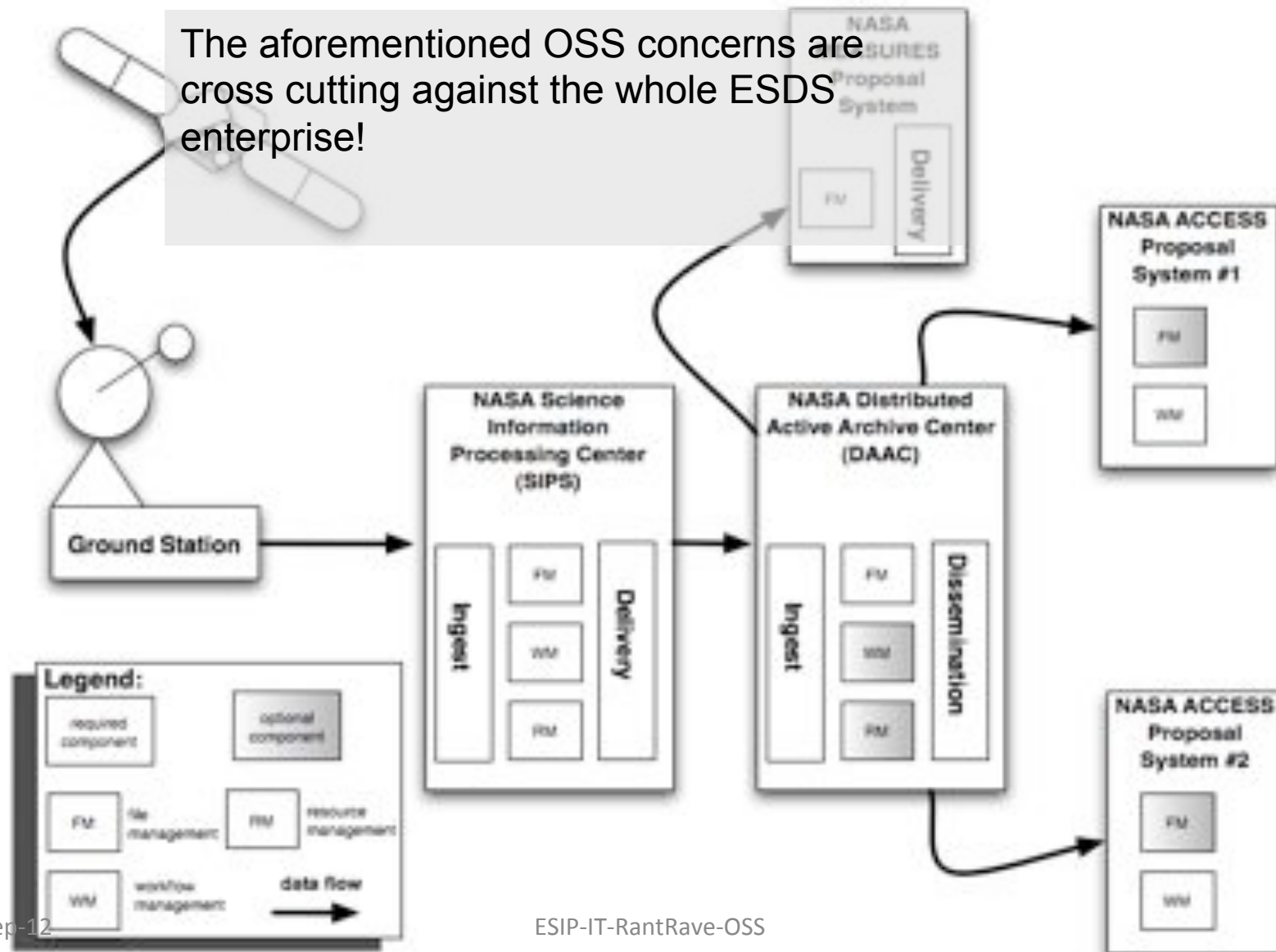


Concerns in the Open Source World

- Help/Guidance
 - Is the OSS community/project alive?
 - How can you tell whether the project is alive?
 - How can you “follow the rainbow” to the OSS pot of gold?
- Interaction
 - What are the best practices for interacting with OSS communities?
- Implementation strategies
 - Insulation: how do you insulate your project from OSS change?
 - Configuration Management: what are the important CM issues when using Open Source Software in your organization?
- Architectural strategies
 - How to design your system to take advantage of OSS?
- Legal strategies
 - How to avoid getting sued by some huge tech company?



NASA Earth Science DS Context





Licensing

- *Relates to: redistribution, intellectual property, contributing, legal strategies*
- There are tons of OSS approved licenses
 - What's the difference between them?
- The difference mostly has to do with
 - Commercialization
 - Redistribution
 - Attribution



Redistribution

- So you've built some awesome piece of Software
- And you're wondering
 - What are my options for distributing it to
 - Other DAACs?
 - Other SIPS?
 - Other funded proposal efforts we're collaborating with (ACCESS, MEASURES)
 - Any other collaborators?
- Question: what license are you going to choose?
 - *Hopefully one that supports redistribution under your own terms!*
- How to redistribute the software?
 - Requires infrastructure
 - Who's going to set it up?



OSS Ecosystems

- Where should you go to for your open source project?



btree lang:c++

Code Results 1 - 10 of about 25,300. (2.62 seconds)

[add/btree.h](#) - 5 identical

```
110: class btree {  
  
114: public:  
    btree(cbdispatch cb);  
    virtual ~btree();
```

[cvs.fu.net/cvs/sfs1](#) - Unknown License - C++

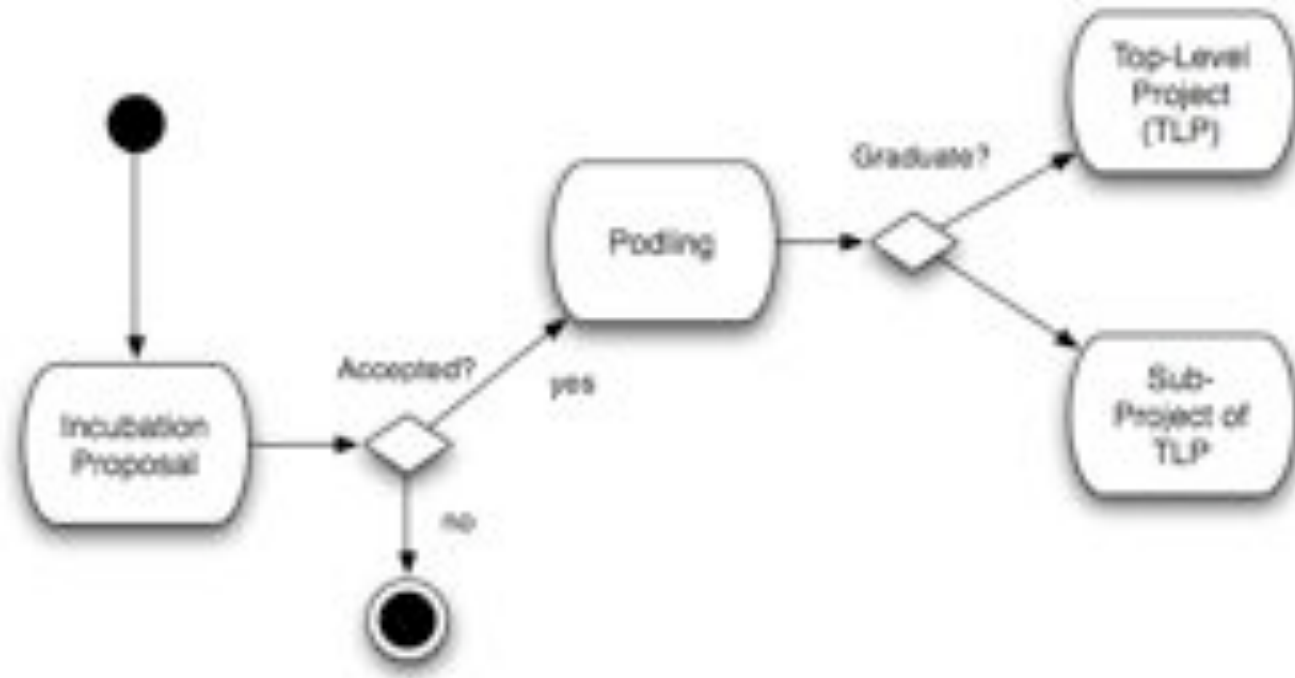


- Should your org have its own?
- Should your project in ESIP have its own?



Apache Maturity Model

- Start out with Incubation
- Grow community
- Make releases
- Gain interest
- Diversify

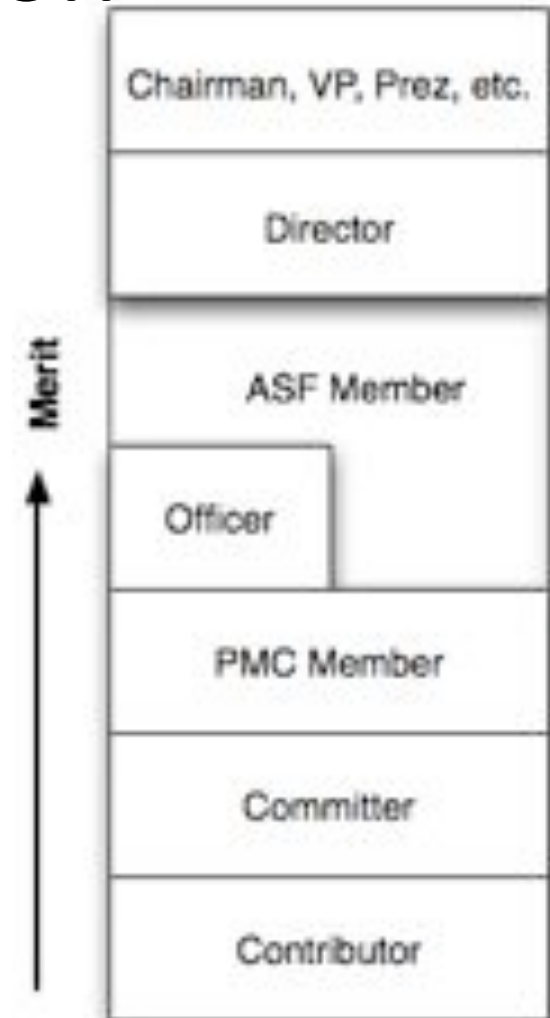


- When the project is ready, graduate into
 - Top-Level Project (TLP)
 - Sub-project of TLP
- Increasingly, Sub-projects are discouraged compared to TLPs



Apache Organization

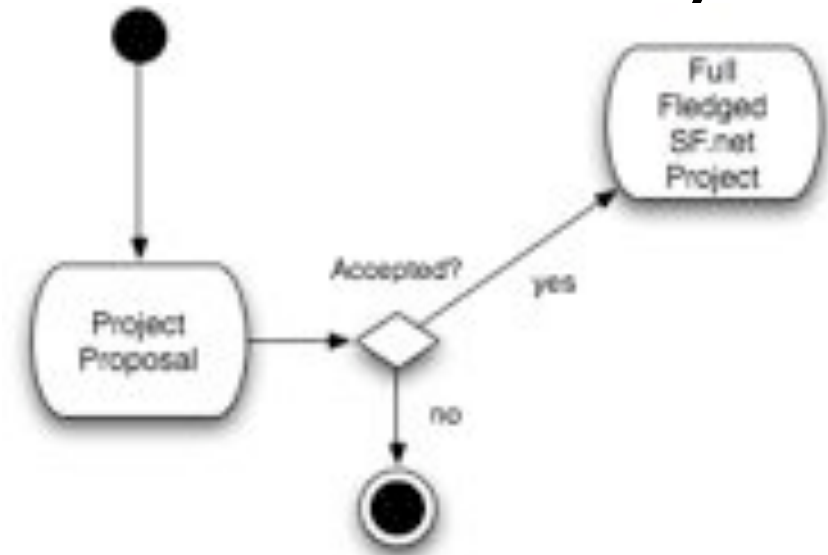
- Apache is a meritocracy
 - You earn your keep and your credentials
- Start out as *Contributor*
 - Patches, mailing list comments, etc.
 - No commit access
- Move onto *Committer*
 - Commit access, evolve the code
- *PMC Members*
 - Have binding VOTES on releases/personnel
- *Officer (VP, Project)*
 - PMC Chair
- *ASF Member*
 - Have binding VOTE in the state of the foundation
 - Elect Board of Directors
- *Director*
 - Oversight of projects, foundation activities





SourceForge (a different model)

- Project Proposal
 - Accepted? Get going!
- No foundation-wide oversight
- Tons of dormant projects with no communities of interest
- Goal is to host infrastructure and host technologies
- Goal is *not* to build communities
- No foundation-wide rules or guidelines for committership or for project management
 - Dealt with locally by the progenitor of the project
 - Can lead to BDFL (benevolent dictator for life) syndrome
- No foundation-wide license requirements
 - BSD, GPL(v2, v3), MIT, LGPL, etc all allowed





OSS Ecosystems

- FTR, there are tons of concerns here
 - Should the ecosystem impose license restrictions?
 - Only support license X or Y?
 - What are the redistribution policies?
 - What's the community?
 - What's the IP?
 - What's the infrastructure support?
 - Is it a foundation with rules, or just free for all?
- Elephant in the room: What is the exposure? How many people are going to community C and are going to see your software and perhaps want to use it, improve it, file bugs against it, file patches, etc.?
- Where/how does this matter to ESIP?
 - Standing up our own OSS ecosystem may make sense for large, coarse-grained federations
 - A LOT more difficult to justify OTOH, for fine grained components, and module reuse



Community versus “Help Desk”

- How do you want to have your open source software project run in the open?
 - Do you expect folks to come to you and *only* file bugs?
 - Then you and your team are the only ones who can fix them?
 - Then you and your team are the only ones who can release updates?
 - → “Help Desk” open source project
 - Examples: Sourceforge.net, Google Code, etc.
 - Do you expect to grow a community of interest where volunteers actively engage in software development?
 - Are volunteers empowered to pick up a shovel and help dig the hole?
 - Can volunteers (including your own paid employees) file issues?
 - Do you want to give the community a stake/vote in the overall process?
 - → “Community Building” open source project
 - Examples: Apache Software Foundation, Eclipse Foundation, etc.



Intellectual Property

- Centers can similarly (if they so choose) have their own customizations/distributions of OSS
 - NASA Langley DAAC's FooBar powered by Apache Hadoop
 - NASA AIRS SIPS's Baz built on Apache Pivot powered by Linux
- NASA protects its marks through the New Technology Report process
 - Uncover marks
 - Uncover potential patents
 - License/etc., as appropriate
 - We see this less on the ground side than we do with the flight
 - Mostly NASA wide, but some center specifics (e.g., Caltech)
- **Takeaway:** the answer to “who owns it” is still “the Center” or “the project” who initiated the software development, with or without OSS. OSS may have its own restrictions/rules, so need to be wary of that (recall *copyleft* syndrome)



Contributing to OSS

- OSS communities are built around “contributions”
 - But what does it mean to contribute?
- Mailing list help/discussions
 - *Yes those are contributions*
- Reporting/filing bugs and new feature improvements
 - *Yes those are contributions*
- Writing documentation and user guides
 - *Yes those are contributions*
- Volunteering to make releases
 - *Yes those are contributions*
- Positively contributing towards the evolution of the project with your thoughts and ideas
 - *Yes those are contributions*
- **NOTE: What did I leave out?**



Contributing to OSS: code

- Yes yes, code is important (but not the only contribution)
- How should you go about your code contributing process to open source?
- *Relates to: “Help Desk” versus Community*
- Do you want the members of your project to be the only folks who can actual touch the source code of your OSS project?
 - Only bug reports?
 - How “open” are you?
 - How “open” do you want to be?
- Do you want to accept code contributions from the community?
 - Patches?
 - Allow community members to become “committers” (assist in the development of the code?)
 - RTC versus CTR



Responsiveness

- Measure of a “healthy” community
- Does your patch sit?
- Do your mailing list questions go unanswered?
- Is the project electing new “committers” (if it’s in community mode)?
- How should your group decide whether or not an OSS project you want to use is responsive?
- How should your group decide how responsive *it will be*?
- Heavily influenced by:
 - Community mode: “Help desk” versus “Community”



Interactions with Open Source

A Bad Email

Subject: Problem with crawling

I am having a problem with crawling the internet. It just seems that it is taking a long time. Does anyone know why crawling takes so long.

Me

A Good Email

Subject: Crawl on 20K pages taking 4 hours

I am using Nutch .8 branch over a cluster of 3 machines each running redhat linux and java 1.5_10 with 500G hard drives, 2.8 Ghz processors, and 2G of ram. I am trying to fetch 20K urls and the fetch process completes fine but when it gets to the reduce process, the cpus go to 100% and the process seems to spin indefinitely. I did a kill -SIGQUIT on the process and it seems to be stuck on the Regex normalizer class. Has anyone experienced similar problems or know what might be causing this problem.

Me

* Taken from: http://wiki.apache.org/nutch/Becoming_A_Nutch_Developer



Leverage active technologies

- Don't expect to use Haskell and have a huge OSS community there to assist you
- Some modern active OSS languages
 - Python
 - Perl
 - Ruby
 - Java
- Don't just be a consumer, be a producer
 - Pick up a shovel and help dig the hole
 - Instead of overlooking the OSS community and telling them how the hole should be dug to meet your needs
 - Will help you get needed OSS feedback for your technology





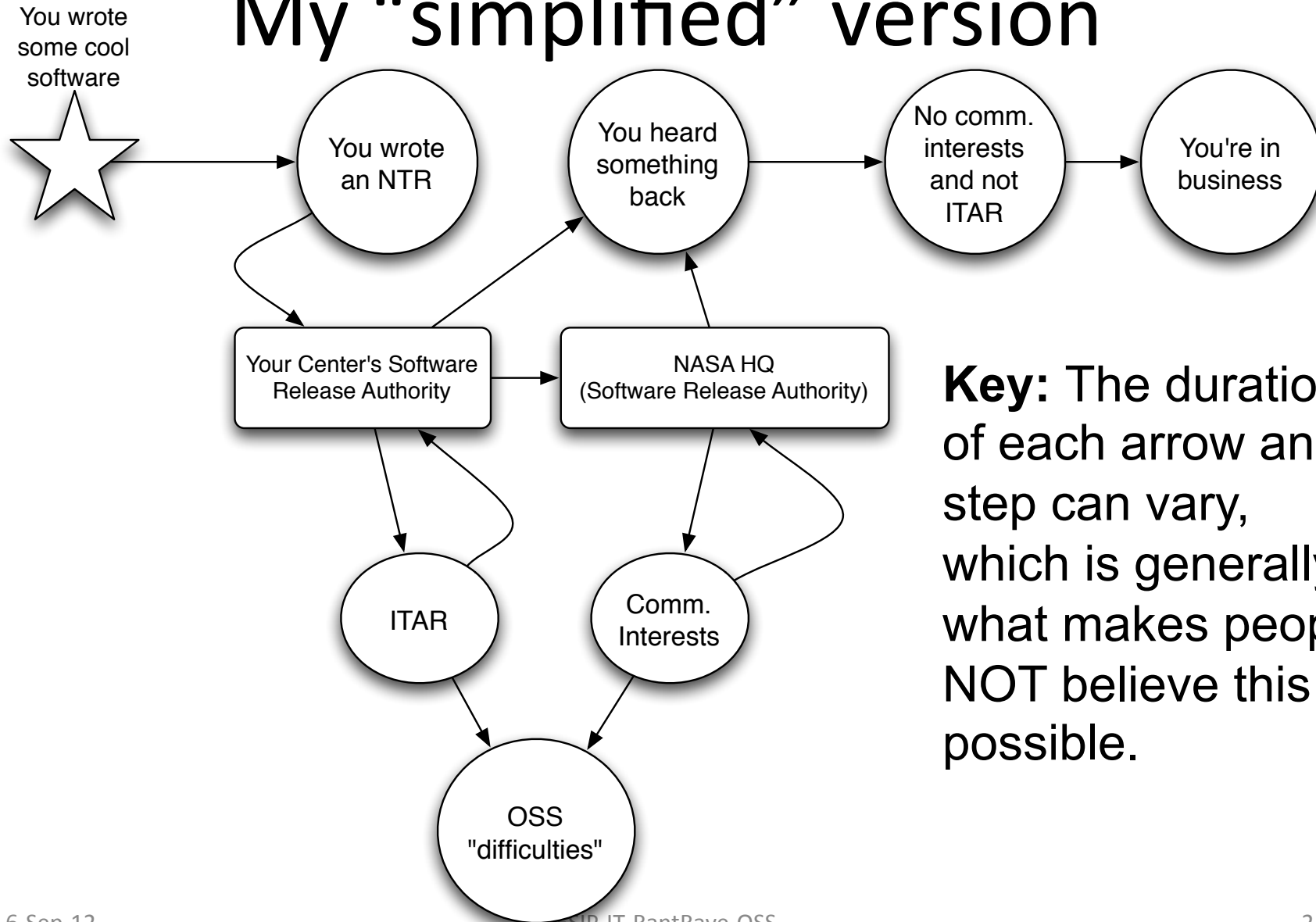
Legal Issues

- Understand patent law
 - Or pay people to understand it for you
- Understand OSS licenses and what you sign up for when you use relevant technologies and their licenses
- Look for open APIs and open specs
 - Understand what “open” means
- Vendor lock in
 - Try to avoid it through the use of all the techniques of OSS which we’ve discussed
- Engage the community and understand what’s “really” free and what isn’t
- Get legal’s buy-in
 - Don’t insulate them



NASA's Process:

My "simplified" version



Key: The duration of each arrow and step can vary, which is generally what makes people NOT believe this is possible.



NASA Open Source Summit



<http://www.nasa.gov/open/source/>



Recap from ESIP Summer 2012

- NASA Open Source “Mini Summit”
 - Presentations from Christine White, Andy Mitchell, Mike Folk and Ken Casey
- ESIP Innovators Talk - Mattmann
- NSF Earth Cube Workflows WG
 - Co-chaired by Yolanda Gil, Chris Mattmann, Paul Ramirez
- ESIP Geospatial Cluster
 - Tommy Jasmin, Paul Ramirez, Cameron Goodale





Recap: NASA OSS 2

- Co-organized and held NASA Open Source Summit 2
 - <http://ossdc.hackpad.com>
- Paul Ramirez and Andrew Hart helped with organization and coordination
- Attendance and presentations by Chris Vein, US Deputy CTO from OSTP along with many other innovators
- Chris Mattmann presentation on Vimeo
<http://vimeo.com/45109553>





ESIP Open Source Cluster

- Discuss aforementioned issues of open source
 - Studies and recommendations
- Real case studies across NASA, NOAA, EPA, other organizations
- Invite members who also *produce* open source software that we consume
- Empower other organizations and individuals with understanding of OSS processes
- Strong ties to NASA ESDSWGs
- Next Meeting
 - 9/12 11am PST/2pm EDT
- Regularly Scheduled Meeting Time
 - 1st Monday of each month, 11am PST/2pm EDT
- WebEx/Conf
 - 1877-668-4493
 - 231 338 97#
 - <http://goo.gl/YEixp>



Wrapup

- Open source is critical to the strategy of the organization
- A great method of *sustainability* and *longevity* of software and community beyond organizational boundaries
- Different licenses, communities, development practices
 - Know what the differences mean



Alright, I'll shut up now

- Any questions?
- THANK YOU!
 - chris.a.mattmann@nasa.gov
 - [@chrismattmann](https://twitter.com/chrismattmann) on Twitter



Backup



Recommendation #1

- **Communication and Publicizing NASA's Open Source Efforts**
 - In order for an Open Source policy to be successful, NASA must make an effort to encourage both internal and external parties to participate in open source development.
 - What does the agency need to do in order to make NASA's open source efforts well known?



Recommendation #2

- **Licensing**

- The NASA Open Source Agreement license (NOSA) was originally developed in 2003 to enable NASA to provide software in source code form to the public, but software must already be considered complete.
- Which licenses should be allowed?



Recommendation #3

- **Barriers to Involvement from the Open Source Community**
 - As a government agency bound by significant regulation and bureaucracy, what are the limits of community contribution For example, could a NASA originated codebase ever be handed over to a non NASA community member for long term support and maintenance? Is this legal? If it were legal, would it be practical?



Recommendation #4

- **Barriers to Development Models and Ongoing Support**
 - How do we ensure that “openness” does not conflict with rigor? How narrow should the definition of “development team” be?



Recommendation #5

- **Government Restrictions**
 - How to we mesh open source software with decidedly un-open policies such as ITAR?



Recommendation #6

- **Limitations on Contributing to External Open Source Projects**
 - What are the differences between contributing minimal, incremental improvements or bug fixes and new features (as per NPR 2210.01)? And, who makes this decision?
 - What lessons learned / best practices can be drawn from current NASA open source development “pathfinder” projects (and can these be applied “generally”)? What should be the policy of NASA personnel contributing in their off-hours? How do you handle / treat situations where people work off-hours on things that derive directly, or are inspired, by what they worked on during duty hours?



Recommendation #7

- **How does Open Source governance look within NASA?**
 - There currently is a lack of information and awareness towards licensing, legal issues, and activity in the open source software community at NASA. How does one receive guidance on open source contributions? What does the process look like?



Recommendation #8

- **How should Open Source Efforts Be Supported?**
 - NASA needs to develop cooperative support into project structure
 - Project Level - permitting code deemed outside the purview of ITAR/EAR to be open source
 - Budget Level - allowing for hiring of floating talent as temporary staff augmentation
 - Organization Level - designing organization to support habits and practices of open source development



Recommendation #9

- **How does NASA open source *everything?***
 - After all, it's taxpayer money, right?



Recommendation #10

- **How to close the feedback loop between policy makers, developers and end users**
 - How do we ensure that draft policies have enough eyes on them, particularly from the people who may be most affected or who have the most detailed knowledge of those areas, and who can thus best understand the implications?



Recommendation #10

- **How to close the feedback loop between policy makers, developers and end users**
 - Policies should not put anyone in the position of performing the mission by violating the policy, or adhering to the policy and thereby reducing the effectiveness or inducing the failure of the mission. Many times, our policies derive (or are simply copied) from Federal law, regulation, guidance or, more commonly, from the policies of other agencies.



Recommendation #10

- **How to close the feedback loop between policy makers, developers and end users**
 - Effective feedback provides opportunities to modify or adjust policy based on practical, realistic feedback. Are we taking advantage of these flexibilities?



Recommendation #11

- **How to encourage cultural change in hiring practices?**
- How can NASA attract more open source savvy people in a world where companies like Red Hat & Google offer careers that encourage such participation? NASA is competing against these companies for the same skills.



Recommendation #12

- **How to Package Open Source Software to be More Accessible?**

- Collaboration on open source software is dependent on others who find the software useful. The barrier for adoption of OSS must be kept low. This also prevents the project dying on the vine. Is there a way we can package OSS to make it easy for others to try and adapt to their needs?
- How does NASA ESDSWG SRWG Packaging Rec fit here?



Recommendation #13

- **Combining open source software development standards with Office of the Chief Engineer Policies**
 - Also interested here in coordinating with the NASA Earth Science Data System WGs, and Software Reuse WG, and ESIP Federation